within the threshold then an undesired change has been made and the process flow proceeds back to block **72**. If the current location is outside the threshold then a desired change has been made and the process flow proceeds to block **76**. By way of example:

Undesired change: |currentUserLocation−lastUserLocation|<Threshold

Desired change: |currentUserLocation−lastUserLocation|≧Threshold

[0062] In one embodiment, the threshold may be defined as the number of sensor levels that need to change in order to report a change in the user finger location to the main system processor of the host device. In one particular implementation, the threshold is equal to about 3. The threshold may be determined by the following equation:

Threshold (T)=C*(native sensor coordinate resolution/logical device unit resolution),

[0063] where the native sensor coordinate resolution defines the maximum number of different positions that the sensors are able to detect for a specific plane coordinate system, the logical device unit resolution defines the number of values that are communicated to the main system processor of the host device for the said specific plane coordinate system, and coefficient C defines the width border area between the clusters of native sensor coordinates that define one logical device unit.

[0064] The coefficient C is generally determined by the sensitivity needed to initiate a user event to the main system processor of the host device. It customizes the threshold value to the physical limitations of the sensor technology and the expected noise of the user finger events. Larger values tend to filter more events and reduce sensitivity. The system designer may pick the exact value of C by testing several values to strike optimal balance between sensitivity and stability of the user finger location. The coefficient C is typically a value between 0 and 0.5, and more particularly about 0.25. As should be appreciated, the threshold (T) is about 2 when the native sensor coordinate resolution is about 1024, the logical device unit resolution is about 128 and the coefficient is about 0.25.

[0065] In block **76**, a new value associated with a particular logical device unit is generated based on the changed native sensor coordinates associated with the particular logical device unit. In most cases, the raw number of slices in the form of native sensor coordinates are grouped into a more logical number of slices in the form of logical device units (e.g., virtual actuation zones).

[0066] Following block **76**, the process flow proceeds to block **78** where the last user location is updated. That is, the last current location is changed to the current user location. The current user location now acts as the last user location for subsequent processing.

[0067] Following block **78**, the process flow proceeds to block **80** where a message is sent. In most cases, the message is sent when the difference between the current and last user location is larger than the threshold value. The message generally includes the new value associated with the selected logical device unit. By way of example, the touch pad may send a message to the main system processor of the host device. When received by the main system processor, the message may be used to make an adjustment in the host device, i.e., cause a control object to move in a specified manner.

[0068] **FIG. 6** is a diagram of a communication protocol **82**, in accordance with one embodiment of the present invention. By way of example, the communication protocol may be used by the user interface and host device of **FIG. 2**. In this particular embodiment, the user interface **22** has one dedicated input ACTIVE line that is controlled by the control circuit **26**. The state of the ACTIVE line signal may be set at LOW or HIGH. The hold switch **28** may be used to change the state of the ACTIVE line signal (for example when the hold switch is in a first position or second position). As shown in **FIG. 6**, when the ACTIVE signal is set to HIGH, the user interface **22** sends a synch message to the control circuit **26** that describes the Button and Touch pad status (e.g., button state and touch pad position). In one embodiment, new synch messages are only sent when the Button state and/or the Touch Pad status changes. For example, when the touch pad position has changed within a desired limit. When the ACTIVE signal is set to LOW, the user interface **22** does not send a synch message to the control circuit **26**. When the ACTIVE signal is toggled from LOW to HIGH, the user interface **22** sends a Button state and touch pad position message. This may be used on startup to initialize the state. When the ACTIVE signal is toggled from HIGH to LOW, the user interface **22** does not send a synch message to the control circuit **26**. In one embodiment, the user interface **22** is configured to send a two data byte message if both the Buttons and touch pad positions changes since the last message was sent, and a one data byte message if only one button state or touch pad position changes.

[0069] **FIG. 7** is a diagram of a message format **86**, in accordance with one embodiment of the present invention. By way of example, the message format **86** may correspond to the synch message described in **FIG. 6**. The message format **86** may form a two data byte message or a one data byte message. Each data byte is configured as an 8 bit message. The upper Most Significant Bit (MSB) of the message is the event type (1 bit) and the lower Least Significant Bits (LSB) are the event value (7 bits).

[0070] The event value is event type specific. In **FIG. 7**, the event type bits are marked as E**0**, and the event value is marked as D**0**-D**6**. As indicated in the diagram, the event type may be a touch pad position change E**1** or a button state change E**0** when the button is being touched or E**1** when the button is not being touched. The event values may correspond to different button events such as seeking forwards (D**4**), seeking backwards (D**3**), playing and pausing (D**2**), providing a menu (D**1**) and making selections (D**0**). The event values may also correspond to touch pad events such as touchpad position (D**5**). For example, in a touch pad that defines the logical coordinates in polar coordinates from 0-127, the event value may correspond to an absolute touch pad position in the range of 0-127 angular positions where zero is 12 o clock, 32 is 3 o clock, 64 is 6 o clock and 96 is 9 o clock, etc. going clockwise. The event values may also correspond to a reserve (D**6**). The reserve is an unused bit that may be used to extend the API.

[0071] **FIG. 8** is a perspective diagram of a media player **100**, in accordance with one embodiment of the present invention. By way of example, the media player **100** may generally correspond to the host device shown in **FIG. 2**. The term "media player" generally refers to computing devices that are dedicated to processing media such as audio, video or other images, as for example, music players, game